68467-US
KK/YO

日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日
Date of Application: ２００３年 ２月２８日

出 願 番 号
Application Number: 特願２００３－０５４０８２

[ST. 10/C]: [ＪＰ２００３－０５４０８２]

出 願 人
Applicant(s): 株式会社デンソー

２００４年 １月２３日

特許庁長官
Commissioner,
Japan Patent Office 今井康夫

出証番号 出証特２００４－３００２２０７

【書類名】　特許願
【整理番号】　PY20030140
【提出日】　平成15年 2月28日
【あて先】　特許庁長官殿
【国際特許分類】　G06F 11/28
【発明者】
　【住所又は居所】　愛知県刈谷市昭和町１丁目１番地　株式会社デンソー内
　【氏名】　岩井　明史
【発明者】
　【住所又は居所】　愛知県刈谷市昭和町１丁目１番地　株式会社デンソー内
　【氏名】　東道　徹也
【特許出願人】
　【識別番号】　000004260
　【氏名又は名称】　株式会社デンソー
【代理人】
　【識別番号】　100068755
　【弁理士】
　【氏名又は名称】　恩田　博宣
【選任した代理人】
　【識別番号】　100105957
　【弁理士】
　【氏名又は名称】　恩田　誠
【手数料の表示】
　【予納台帳番号】　002956
　【納付金額】　21,000円
【提出物件の目録】
　【物件名】　明細書　1
　【物件名】　図面　1
　【物件名】　要約書　1

【包括委任状番号】　9908214

【プルーフの要否】　　要

【書類名】　明細書

【発明の名称】　制御プログラムの作成方法及び検査プログラム

【特許請求の範囲】

【請求項１】制御要求仕様が要求仕様記述言語にて記述された制御モデルから所定のプログラム言語にて記述されたコードを自動生成する自動コード生成手段を用いて該所定のプログラム言語にて記述された制御プログラムを作成する制御プログラムの作成方法において、

前記制御モデル及び前記制御プログラムの少なくとも一方の異常の有無を検査するに際し、前記制御モデルの挙動をシミュレーションするシミュレーション手段の動作と前記制御プログラムを実行するプログラム実行手段の動作とを同期させる同期手段を用いる

ことを特徴とする制御プログラムの作成方法。

【請求項２】前記同期手段は、前記自動コード生成手段によるコードの自動生成に際して取得される前記制御モデルと該制御モデルから生成されるコードとの対応関係を表す対応情報に基づいて前記シミュレーション手段の動作と前記実行手段の動作とを同期させるものである

請求項１記載の制御プログラムの作成方法。

【請求項３】前記同期手段による前記シミュレーション手段の動作と前記プログラム実行手段の動作との同期が、前記制御モデル及び前記制御プログラムのいずれか一方に対して動作の停止箇所を指定するブレークポイントが設定されたときに他方に対しても前記対応情報に基づきその対応するブレークポイントを設定することによって行われる

請求項２記載の制御プログラムの作成方法。

【請求項４】前記ブレークポイントが、前記制御モデルを構成する各機能ブロック毎に設定可能とされる

請求項２又は３記載の制御プログラムの作成方法。

【請求項５】前記同期手段による前記シミュレーション手段の動作と前記プログラム実行手段の動作との同期が、前記プログラム実行手段により前記制御プログラムを１行ずつ実行及び停止させることによって行われる

請求項２記載の制御プログラムの作成方法。

【請求項６】請求項２～５のいずれかに記載の制御プログラムの作成方法において、

前記異常の有無の検査が、前記同期手段によって互いに同期された前記制御モデルのシミュレーション結果と前記制御プログラムの実行結果とを逐次比較する比較手段によって行われる

ことを特徴とする制御プログラムの作成方法。

【請求項７】前記制御モデルのシミュレーション結果と前記制御プログラムの実行結果との逐次比較が、前記対応情報に基づく前記制御モデルのシミュレーション順序と前記制御プログラムの実行順序との比較として行われる

請求項６記載の制御プログラムの作成方法。

【請求項８】前記制御モデルのシミュレーション結果と前記制御プログラムの実行結果との逐次比較が、前記対応情報に基づく前記制御モデルのシミュレーションによって生成される値と前記制御プログラムの実行にかかる変数値との比較として行われる

請求項６又は７記載の制御プログラムの作成方法。

【請求項９】前記対応情報に基づく前記制御モデルのシミュレーションによって生成される値と前記制御プログラムの実行にかかる変数値との比較による異常の有無の検査が、前記制御モデルのシミュレーションによって生成される値と前記制御プログラムの実行にかかる変数値との差が許容範囲内にないか否かによって行われる

請求項８記載の制御プログラムの作成方法。

【請求項１０】請求項６～９のいずれかに記載の制御プログラムの作成方法において、

前記比較手段により異常がある旨の判断がなされたとき、その時点での前記制御モデルのシミュレーション箇所と前記制御プログラムの実行箇所とを出力するようにした

ことを特徴とする制御プログラムの作成方法。

【請求項１１】請求項３～１０のいずれかに記載の制御プログラムの作成方

法において、

前記同期手段による前記制御モデルのシミュレーション及び前記制御プログラムの実行の一旦停止時、前記シミュレーションの停止箇所において保持されるパラメータ及び前記実行の停止箇所で保持される変数値の少なくとも一方を変更可能とする

ことを特徴とする制御プログラムの作成方法。

【請求項１２】制御要求仕様が要求仕様記述言語にて記述された制御モデルから所定のプログラム言語にて記述されたコードを自動生成する自動コード生成手段を用いて該所定のプログラム言語にて記述された制御プログラムを作成する際に用いられる検査プログラムであって、

前記制御モデル及び前記制御プログラムの少なくとも一方の異常の有無を検査すべくコンピュータを通じて実行される処理として、

ａ．前記コードを自動生成する際に取得される情報から前記制御モデルと前記制御プログラムとの対応関係を表す対応情報を生成し、前記制御モデルをシミュレーションするシミュレーション手段と前記制御プログラムを実行するプログラム実行手段との双方の動作の停止箇所を前記対応情報に基づき対応付けする処理、

ｂ．前記シミュレーション手段と前記プログラム実行手段との双方に対してシミュレーション及びプログラムの実行を指示する処理、

ｃ．前記シミュレーション及びプログラムの実行の後、前記シミュレーション手段及び前記プログラム実行手段の双方の停止を検出する処理、

ｄ．前記双方の停止を検出したとき、前記制御モデルのシミュレーションの結果と前記制御プログラムの実行の結果とを比較し、その比較結果に基づいて前記異常の有無を検査する処理、

を含むことを特徴とする検査プログラム。

【請求項１３】前記対応情報は、前記制御モデルと前記制御プログラムとの互いに対応する制御位置の関係を表す情報である実行位置対応情報を有して生成され、

前記停止箇所を前記対応情報に基づき対応付けする処理が、前記シミュレーション手段及び前記プログラム実行手段のいずれか一方においてその動作の停止箇

所を指定するブレークポイントが設定されることによって前記実行位置対応情報に基づき他方の対応する箇所に対応するブレークポイントを設定する処理からなる

請求項１２記載の検査プログラム。

【請求項１４】前記異常の有無を検査する処理が、前記制御モデルのシミュレーションの停止位置と前記制御プログラムの実行の停止位置とを前記実行位置対応情報に基づき比較することによって行われる

請求項１３記載の検査プログラム。

【請求項１５】前記対応情報は、前記制御モデルの処理にかかるパラメータと前記制御プログラムの処理にかかる変数値との対応関係を表す変数対応情報を更に含み、

前記異常の有無を検査する処理が、前記制御モデルのシミュレーションによって生成される値と前記制御プログラムの実行にかかる変数値とを前記変数対応情報に基づき比較することによって行われる

請求項１２～１４のいずれかに記載の検査プログラム。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は、制御対象の制御に用いる制御プログラムの作成方法及び該制御プログラムの作成に際して用いられる検査プログラムに関する。

【０００２】

【従来の技術】

例えば車載エンジン等の運転を制御する制御プログラムの開発、作成には通常、

（ア）設計（制御）仕様に基づいて制御プログラムを作成する工程。

（イ）該作成した制御プログラムの異常の有無を検査、修正（デバッグ）する工程。

（ウ）該デバッグした制御プログラムを用いて実地試験を行う工程。

（エ）この実地試験に基づいて更に制御プログラムを修正する工程。

等々が含まれる。

【０００３】

また近年は、こうした制御プログラムの開発、作成において、その要求仕様を記述する制御モデルからプログラム言語にて記述されたコードを自動的に生成する工程を有するモデルベース開発も行われている。このモデルベース開発を用いる場合、上記（ア）の工程は通常、コードを自動生成する工程に加えて、自動生成されたコードに適宜のプログラムを追加するなどの工程も含まれることとなる。

【０００４】

ここで一般に、上記モデルベース開発では、制御モデルを適宜のハードウェアにてシミュレーションすることで、その妥当性が検証される。こうした検証のなされた制御モデルから自動生成されたコードは完成度が高いため、上記（エ）の工程における制御プログラムの修正を低減することができるようになる。また、制御モデルの構築自体は、制御プログラムの構築よりも簡易に行うことができるため、こうしたモデルベース開発を採用することで、制御プログラムの開発、作成の簡易化を図ることも可能となる。

【０００５】

【発明が解決しようとする課題】

ところで、上記自動生成されたコードには、制御モデルの指定した各処理について、同制御モデルには含まれていなかった実行順序が新たに生成され、追加されることがある。また通常、自動生成された各コード間の優先順位を規定するプログラムは自動生成されないため、上記制御プログラムにはこうした優先順位を規定するためのプログラムを新たに追加する必要もある。

【０００６】

このため、上記モデルベース開発を採用する場合、制御モデルから自動生成されるコードに基づく制御プログラムのデバッグ作業や、上記制御モデルがコードを自動生成するのに妥当なものであるか否かを検査し修正するデバッグ作業が逆に困難なものとなる。これは、上述のように制御モデルと制御プログラムとの間の実行順序が必ずしも自明でないことに起因するものであり、制御モデル及び制

御プログラムの一方のデバッグ作業にて何らかの問題が発見されたときには、これと対応する他方の実行条件を特定することが極めて困難となる。

【０００７】

具体的には、例えば自動生成されたコードの有する全ての分岐処理を網羅するようなデバッグを行う場合には、同一のテストデータを用いて制御モデルのシミュレーション及び制御プログラムの実行を行い、これらシミュレーション結果と制御プログラムの実行結果とを比較する必要がある。しかし、制御モデルのシミュレーション結果と制御プログラムでの分岐処理とを対応させること自体が困難であることから、これら各分岐処理において双方の処理結果を正確に比較することは非常に困難である。

【０００８】

また、例えば制御プログラムのデバッグにおいて、所定の箇所に割り込み禁止処理や異常処理等を新たに追加することが望まれる場合、制御プログラムのこうした所定の箇所と対応する箇所が制御モデルのいずれの箇所にあたるかを特定することも極めて困難である。このため、上記所定箇所に割り込み禁止処理や異常処理等を追加することについては、その妥当性を判断すること自体が困難なものとなっている。

【０００９】

本発明は、こうした実情に鑑みてなされたものであり、その目的は、モデルベース開発を行う場合にあって、制御プログラムや制御モデルの異常の有無の検査を簡易に行うことのできる制御プログラムの作成方法と該制御プログラムの作成に用いる検査プログラムとを提供することにある。

【００１０】

【課題を解決するための手段】

こうした目的を達成すべく、請求項１記載の制御プログラムの作成方法では、制御モデル及び制御プログラムの少なくとも一方の異常の有無を検査するに際し、前記制御モデルの挙動をシミュレーションするシミュレーション手段の動作と前記制御プログラムを実行するプログラム実行手段の動作とを同期させる同期手段を用いるようにした。

【００１１】

このように制御プログラムをプログラム実行手段によって実行することでその異常の有無の検査を行うに際し、これを制御モデルのシミュレーションと同期させるようにすることで、これら制御モデルのシミュレーション状況と制御プログラムの実行状況とを簡易に比較することができるようになる。

【００１２】

このため、制御モデルが規定する各処理について同制御モデルの有しない実行順序が新たに追加されることがあるコードを用いて作成された制御プログラムと、制御モデルとの対応関係を簡易に把握することができるようになる。したがって、モデルベース開発を行う場合にあって、制御プログラムや制御モデルの異常の有無の検査を簡易に行うことができるようになる。

【００１３】

なお、上記プログラム実行手段にて実行される制御プログラムは、自動生成されるコードそのものでもよく、また、自動生成される複数のコード間の優先順位等を規定するプログラム等が追加されたものなどでもよい。

【００１４】

また、請求項２記載の制御プログラムの作成方法では、同期手段として、前記自動コード生成手段によるコードの自動生成に際して取得される前記制御モデルと該制御モデルから生成されるコードとの対応関係を表す対応情報に基づいて前記シミュレーション手段の動作と前記実行手段の動作とを同期させるものを用いた。

【００１５】

制御モデルからコードを自動生成する自動コード生成手段は、同自動生成に際し制御モデルとコードとの対応関係を表す情報を有することとなる。そして、この対応情報を用いることで、シミュレーション手段による制御モデルのシミュレーションの実行位置とプログラム実行手段による制御プログラムの実行位置とを適切に対応付けることができる。

【００１６】

また、請求項３記載の制御プログラムの作成方法では、同期手段による前記シ

ミュレーション手段の動作と前記プログラム実行手段の動作との同期が、前記制御モデル及び前記制御プログラムのいずれか一方に対して動作の停止箇所を指定するブレークポイントが設定されたときに他方に対しても前記対応情報に基づきその対応するブレークポイントを設定することによって行われるようにした。

【００１７】

シミュレーション手段やプログラム実行手段は、それぞれのブレークポイントでシミュレーションの実行や制御プログラムの実行を停止する。このため、対応情報により把握される制御モデルと制御プログラム（コード）との対応関係に基づき、それぞれの対応する実行位置にブレークポイントを設定することで、制御モデル及び制御プログラムについての上記実行位置が互いに対応する位置となる。したがって、これらシミュレーション手段とプログラム実行手段との実行、停止を互いに同期させることができるようになる。

【００１８】

更に、上記のように制御モデル及び制御プログラムのいずれか一方にブレークポイントが設定されたときに、同期手段により他方にもブレークポイントが自動的に設定されるようにした。このため、例えば、外部からユーザが制御モデル及び制御プログラムのうちのいずれか一方を任意に選択してブレークポイントを設定することで、双方の適切な箇所にブレークポイントを設定することができる。

【００１９】

また、請求項４記載の制御プログラムの作成方法では、ブレークポイントを、前記制御モデルを構成する各機能ブロック毎に設定可能とした。

制御モデルから自動生成されるコードには、制御モデルの各機能ブロックに対応する処理について同制御モデルには規定されていない実行順序が新たに追加されることがある。このため、自動生成されたコードの検査においては、これら新たに追加される実行順序の異常の有無を検査することが望ましい。

【００２０】

この点、ブレークポイントを各機能ブロックに設定可能とすることで、新たに実行順序が追加される場合であれ、その異常の有無を、制御モデルと対応させつつ簡易に検査することができるようになる。

【００２１】

また、請求項５記載の制御プログラムの作成方法では、前記同期手段による前記シミュレーション手段の動作と前記プログラム実行手段の動作との同期が、前記プログラム実行手段により前記制御プログラムを１行ずつ実行及び停止させることによって行われるようにした。

【００２２】

このように制御プログラムを１行ずつ実行及び停止させるようにすることで、停止箇所の設定を簡易に行うことができる。

また、請求項６記載の制御プログラムの作成方法では、前記異常の有無の検査が、前記同期手段によって互いに同期された前記制御モデルのシミュレーション結果と前記制御プログラムの実行結果とを逐次比較する比較手段によって行われるようにした。

【００２３】

このように、制御モデル及び制御プログラムの実行結果を逐次比較する比較手段を用いることで、これら制御モデルのシミュレーション結果と制御プログラムの実行結果との対応関係を自動的に検出することができるようになる。したがって、上記異常の有無の検査を自動的に行うことができる。

【００２４】

また、請求項７記載の制御プログラムの作成方法では、前記制御モデルのシミュレーション結果と前記制御プログラムの実行結果との逐次比較が、前記対応情報に基づく前記制御モデルのシミュレーション順序と前記制御プログラムの実行順序との比較として行われるようにした。

【００２５】

自動生成されるコードには、制御モデルの処理について同制御モデルが有しない実行順序が新たに追加されることがある。このため、同コードを用いて作成される制御プログラムと制御モデルとは互いの処理の実行順序が異なったものとなることがある。

【００２６】

この点、上記のように比較手段によって制御モデルのシミュレーションの順序

と制御プログラムの実行順序とを比較することで、この実行順序の不一致にかかる異常を自動的に検出することができるようになる。

【００２７】

また、請求項８記載の制御プログラムの作成方法では、前記制御モデルのシミュレーション結果と前記制御プログラムの実行結果との逐次比較が、前記対応情報に基づく前記制御モデルのシミュレーションによって生成される値と前記制御プログラムの実行にかかる変数値との比較として行われるようにした。

【００２８】

自動生成されたコードが適切なものでないと、コード（制御プログラム）の実行によって演算される変数の値は、制御モデルの対応する処理によって生成される値と異なることがある。

【００２９】

この点、上記のように比較手段によって制御モデルのシミュレーションによって生成される値と制御プログラムの実行にかかる変数値とを比較することで、これらの不一致にかかる異常を自動的に検出することができるようになる。

【００３０】

なお、こうした変数値の比較による異常の有無の検査は、請求項９記載の制御プログラムの作成方法によるように、前記対応情報に基づく前記制御モデルのシミュレーションによって生成される値と前記制御プログラムの実行にかかる変数値との比較による異常の有無の検査が、前記制御モデルのシミュレーションによって生成される値と前記制御プログラムの実行にかかる変数値との差が許容範囲内にないか否かによって行われるにしてもよい。

【００３１】

これにより、異常の有無の検出をより好適に行うことができるようになる。

また、請求項１０記載の制御プログラムの作成方法では、前記比較手段により異常がある旨の判断がなされたとき、その時点での前記制御モデルのシミュレーション箇所と前記制御プログラムの実行箇所とを出力するようにした。

【００３２】

このように制御モデルのシミュレーション箇所と前記制御プログラムの実行箇

所とを検出し、出力することで、異常がある旨判断された箇所を、外部のユーザが簡易に把握することができるようになる。

【００３３】

また、請求項１１記載の制御プログラムの作成方法では、前記同期手段による前記制御モデルのシミュレーション及び前記制御プログラムの実行の一旦停止時、前記シミュレーションの停止箇所において保持されるパラメータ及び前記実行の停止箇所で保持される変数値の少なくとも一方を変更可能とした。

【００３４】

このようにシミュレーションの停止箇所において保持されるパラメータ及び実行の停止箇所で保持される変数値の少なくとも一方を変更可能とすることで、異常がある旨判断がなされた場合であれ、再度制御モデルのシミュレーション条件と制御プログラムの実行条件とを等しくすることができる。このため、異常がある旨判断された箇所から異常の有無の検査を再開することが可能となる。

【００３５】

また、請求項１２記載の検査プログラムでは、前記制御モデル及び前記制御プログラムの少なくとも一方の異常の有無を検査すべくコンピュータを通じて実行される処理として、ａ．前記コードを自動生成する際に取得される情報から前記制御モデルと前記制御プログラムとの対応関係を表す対応情報を生成し、前記制御モデルをシミュレーションするシミュレーション手段と前記制御プログラムを実行するプログラム実行手段との双方の動作の停止箇所を前記対応情報に基づき対応付けする処理、ｂ．前記シミュレーション手段と前記プログラム実行手段との双方に対してシミュレーション及びプログラムの実行を指示する処理、ｃ．前記シミュレーション及びプログラムの実行の後、前記シミュレーション手段及び前記プログラム実行手段の双方の停止を検出する処理、ｄ．前記双方の停止を検出したとき、前記制御モデルのシミュレーションの結果と前記制御プログラムの実行の結果とを比較し、その比較結果に基づいて前記異常の有無を検査する処理、を含むようにした。

【００３６】

このようにシミュレーション手段とプログラム実行手段との双方の停止箇所を

対応付けることで、シミュレーション手段とプログラム実行手段とを同期させることができるようになる。そして、これらシミュレーション手段とプログラム実行手段とのそれぞれに対し、実行を指示するとともに、双方の停止を検出したときにシミュレーションの結果と制御プログラムの実行の結果とを比較することで、双方の停止位置において異常の有無を自動的に検査することができるようになる。

【００３７】

また、請求項１３記載の検査プログラムでは、前記対応情報は、前記制御モデルと前記制御プログラムとの互いに対応する制御位置の関係を表す情報である実行位置対応情報を有して生成され、前記停止箇所を前記対応情報に基づき対応付けする処理が、前記シミュレーション手段及び前記プログラム実行手段のいずれか一方においてその動作の停止箇所を指定するブレークポイントが設定されることによって前記実行位置対応情報に基づき他方の対応する箇所に対応するブレークポイントを設定する処理からなる。

【００３８】

上記のようにブレークポイントを自動的に設定することで、例えば、外部からユーザが制御モデル及び制御プログラムのうちのいずれか一方を任意に選択してブレークポイントを設定すること、双方の適切な箇所にブレークポイントを設定することができる。

【００３９】

また、請求項１４記載の検査プログラムでは、前記異常の有無を検査する処理が、前記制御モデルのシミュレーションの停止位置と前記制御プログラムの実行の停止位置とを前記実行位置対応情報に基づき比較することによって行われる。

【００４０】

自動生成されるコードには、制御モデルの処理についての実行順序であって同制御モデルが有しない実行順序が新たに追加されることがある。このため、同コードを用いて作成される制御プログラムと制御モデルとは互いの処理の実行順序が異なったものとなることがある。

【００４１】

この点、上記のように制御モデルの実行順序と制御プログラムの実行順序とを比較することで、この実行順序の不一致にかかる異常を自動的に検出することができるようになる。

【００４２】

また、請求項１５記載の検査プログラムでは、前記対応情報は、制御モデルの処理にかかるパラメータと前記制御プログラムの処理にかかる変数値との対応関係を表す変数対応情報を更に含み、前記異常の有無を検査する処理が、前記制御モデルのシミュレーションによって生成される値と前記制御プログラムの実行にかかる変数値とを前記変数対応情報に基づき比較することによって行われる。

【００４３】

自動生成されたコードが適切なものでないと、コード（制御プログラム）の実行によって演算される変数の値は、制御モデルの対応する処理によって生成される値と異なることがある。

【００４４】

この点、制御モデルのシミュレーションによって生成される値と制御プログラムの実行にかかる変数値とを比較することで、これらの不一致にかかる異常を自動的に検出することができるようになる。

【００４５】

【発明の実施の形態】

以下、本発明にかかる制御プログラムの作成方法を車両の制御プログラムの作成方法に適用した一実施形態を図面を参照しつつ説明する。

【００４６】

図１は、本実施形態にかかる制御プログラムの作成手順の概要を示すフローチャートである。

同図１に示すステップＳ１では、まず、作成対象となる制御プログラムについて、その要求仕様を記述する制御モデルを要求仕様記述言語を用いて構築する。次に、こうして構築された制御モデルをシミュレーションすることで、同制御モデルが正当な機能と妥当な性能を有するものであるか否かを検証する。そして、この検証結果に基づき、適宜、制御モデルを修正する。

【００４７】

次に、ステップＳ２において、制御モデルからＣ言語にて記述されたコードを自動生成する。このコードには、制御モデルの規定する処理について同制御モデルによって定義されていない実行順序が新たに定義されている。そして、このコードに、更に各コード間の優先順位を定義するプログラム等、制御モデルを用いることなくＣ言語を用いて直接構築されたプログラムが追加されることで制御プログラムが生成される。

【００４８】

ステップＳ３では、上記制御モデルをシミュレーションするとともに、制御プログラムを実行することで、互いの対応関係を維持しつつ制御モデルや制御プログラムの異常の有無の検査を行う。そして、制御プログラムに異常がある旨判断されると、制御プログラムが修正される。また、制御モデルから妥当なコードを自動生成することが困難であると判断される場合には、上記ステップＳ１に戻り、制御モデルを修正するようにしてもよい。ちなみに、ここで、上記制御プログラムの実行は、図２に示す態様にて行われる。同図２に示すように、コンピュータ１には、リアルタイムオペレーティングシステム（ＲＴＯＳ）２が搭載されている。そして、同ＲＴＯＳ２上で、上記Ｃ言語を用いて直接記述されたプログラムである追加プログラム３やコード４からなる制御プログラムが実行される。

【００４９】

こうして制御プログラムのデバッグが終了すると、先の図１に示すステップＳ４において、同制御プログラムを搭載した電子制御装置によって実際の車両の走行を制御する実地試験を行う。そして、この実地試験に基づき制御プログラムに妥当でない箇所がある場合には、ステップＳ１に戻って制御モデルを修正し、上記ステップＳ１～Ｓ４の処理を繰り返す。この一連の処理は、制御プログラムが妥当であると判断されるまで行われる。

【００５０】

ここで、上記ステップＳ３の処理である制御プログラムのデバッグ作業について、詳細に説明する。

図３は、制御プログラムの作成及びデバッグに用いるシステムの全体構成を示

す図である。

【００５１】

モデル格納部１０には、先の図１のステップＳ１にて構築され検証のなされた制御モデルが格納されている。図４に、こうした制御モデルの一例を模式的に示す。

【００５２】

同図４に示す制御モデルは、車載エンジンの燃料噴射量の演算、及び噴射タイミングの演算を行うモデルである。ブロックＢ１は、所定のクランク角をトリガとする機能を示す。ブロックＢ２は、上記所定のクランク角をトリガとして基本噴射量を演算する機能を示す。ブロックＢ３は、上記ブロックＢ２によって演算された基本噴射量をブロックＢ４に書き込む機能を示す。ブロックＢ４は、基本噴射量を記憶する機能を示す。ブロックＢ５は、記憶されている基本噴射量を読み出す機能を示す。ブロックＢ６は、上記所定のクランク角をトリガとして上記基本燃料噴射量に対する補正量を計算する機能を示す。ブロックＢ７は、補正量と上記読み出された基本噴射量とに基づき噴射タイミングを演算する機能を有する。

【００５３】

ちなみに、図４において実線で示す矢印は、処理の順序を規定するとともに、各処理の施されたデータの流れを示すものである。すなわち例えば、ブロックＢ７の処理は、ブロックＢ５及びブロックＢ６の処理の後に行われるようにその処理順序が規定されている。しかし、ブロックＢ２とブロックＢ６との処理については、その順序は規定されていない。また、ブロックＢ３及びブロックＢ５の処理の順序についても規定されていない。

【００５４】

一方、図３に示す自動コード生成手段１２は、上記制御モデルを入力とし、これから上記Ｃ言語で記述されたコードを生成して出力する。プログラム格納部１４には、生成されたコードと先の図２に示した追加プログラム３とからなる制御プログラムが格納されている。

【００５５】

図５に、先の図４に示す制御モデルから自動生成されるコードを例示する。同図５に示すように、先の図４に示す制御モデルにおけるブロックＢ３の処理はブロックＢ５の処理に先立ち行われる必要があるにもかかわらず、このプログラムでは、ブロックＢ５に対応する処理の後にブロックＢ３に対応する処理が行われる。これは制御モデルには、各機能ブロックの示す処理の実行順序が必ずしも定められておらず、自動コード生成手段１２によってこの実行順序が新たに定められることに起因している。すなわち、制御モデルにはブロックＢ５とブロックＢ３との処理の順序が規定されていないのであり、これらの処理の実行順序が自動コード生成手段１２によって新たに追加される際、適切なものとならなかった場合を図５は示している。

【００５６】

また、対応情報作成手段１６は、上記自動コード生成手段１２がコードを生成するに際して同自動コード生成手段１２の有する情報に基づき、制御モデルとコードとの対応関係を表す対応情報を作成する。この対応情報は、制御モデルのブロックとコードの実行ポイントとの対応関係を示す実行位置対応情報と、制御モデルのブロック結線や、ストレージ（Ｂ４等、記憶にかかるブロック）とコードの変数との対応関係を示す変数対応情報とからなる。

【００５７】

図６に、これら実行位置対応情報と変数対応情報とからなる対応情報を例示する。同図６においては、実行位置対応情報として、制御モデルのブロックと、コードの有するラベルとが対応付けられている。また、変数対応情報として、制御モデルのブロック結線／ストレージとコードの変数とが対応付けられている。

【００５８】

また、対応情報格納部１８は、上記対応情報を格納する。

シミュレーション手段２０は、制御モデルを入力とし、その動作をシミュレーションする。また、プログラム実行手段２２は、制御プログラムを入力とし、これを実行するものである。なお、このプログラム実行手段２２は、先の図２に示したコンピュータ１及びＲＴＯＳ２を備えて構成される。

【００５９】

また、ブレークポイント設定手段２４は、外部からの入力に基づき制御モデル及び制御プログラムのいずれかに動作の停止箇所を指定するブレークポイントの設定をシミュレーション手段２０及びプログラム実行手段２２に指示する。このブレークポイントは、制御モデルの各ブロックに設定可能とする。また、このブレークポイントは、例えば制御プログラムの上記ラベルを用いて指定される各実行ポイントに設定可能とする。

【００６０】

なお、このブレークポイント設定手段２４のうち制御モデルにブレークポイントを設定する機能は上記シミュレーション手段２０に備えられるようにしてもよい。また、ブレークポイント設定手段２４のうち制御プログラムにブレークポイントを設定する機能は上記プログラム実行手段２２に備えられるようにしてもよい。

【００６１】

同期手段２６は、ブレークポイント設定手段２４を通じて制御モデル及び制御プログラムのいずれか一方に、その動作の停止箇所を指定するブレークポイントが設定されたとき、上記対応情報に基づき他方に対し対応する動作の停止箇所を指定するブレークポイントを設定する。すなわち、上記対応情報のうち実行位置対応情報を用いて、制御モデルのブロックから対応するコードの実行ポイントを検索することで、又は、コードの実行ポイントから制御モデルのブロックを検索することで、ブレークポイントを設定する。

【００６２】

更に、同期手段２６は、シミュレーション手段２０及びプログラム実行手段２２の動作が双方ともブレークポイントにて一旦停止したときに、これらシミュレーション結果及び実行結果を比較する比較手段を備えている。

【００６３】

表示手段２８は、上記比較手段による比較結果により不一致が検出され、制御モデル及び制御プログラムのいずれかに問題があるときなどにこれを表示するものである。結果格納部３０は、比較手段の比較結果を格納する。

【００６４】

ちなみに、上記モデル格納部１０、プログラム格納部１４、対応情報格納部１８、結果格納部３０は、例えばハードディスク装置等の記憶装置によって構成される。また、自動コード生成手段１２、対応情報作成手段１６、シミュレーション手段２０、プログラム実行手段２２、ブレークポイント設定手段２４、同期手段２６は、処理手順に関するプログラムが記録されたハードディスク装置やＲＯＭ等の記憶装置とコンピュータとを備えて構成される。

【００６５】

以下、こうした構成を有するシステムにおける処理である制御プログラムの作成及びデバッグにかかる処理について詳細に説明する。

本実施形態では、モデル格納部１０に格納された制御モデルをシミュレーションするシミュレーション手段２０とプログラム格納部１４に格納された制御プログラムを実行するプログラム実行手段２２との動作を同期手段２６によって同期させる。そして、上述したブレークポイントでこれらシミュレーション手段２０とプログラム実行手段２２との動作が停止したとき、上記比較手段によってシミュレーションの結果とプログラムの実行結果とを比較することで、制御モデルや制御プログラムの異常の有無を検査する。そして、この検査の結果、制御プログラムに問題があるときには、又は、制御モデルが妥当なコードを生成することができないときには、制御プログラムや制御モデルを修正する。

【００６６】

ここでまず、上記ブレークポイント設定手段２４及び同期手段２６によるブレークポイントの設定にかかる処理について図７を用いて説明する。

同図７は、同期手段２６によるブレークポイントの設定にかかる処理の手順を示すフローチャートである。

【００６７】

この一連の処理においては、ステップＳ３０１及びステップＳ３０２において、上記ブレークポイント設定手段２４を通じて上記シミュレーション手段２０及び上記プログラム実行手段２２のいずれかからブレークポイントが設定されたか否かを判断する。すなわち、同期手段２６では、シミュレーション手段２０及びプログラム実行手段２２にアクセスすることで、ブレークポイントの設定がなさ

れたか否かを判断する。

【００６８】

そして、ステップＳ３０３では、実行位置対応情報に基づいて制御モデル上に設定されたブレークポイントに該当する制御プログラム上の位置を決定するとともに、この決定された位置をブレークポイントとして制御プログラム上に設定するようにプログラム実行手段２２に指示する。

【００６９】

また、ステップＳ３０４では、実行位置対応情報に基づいて制御プログラム上に設定されたブレークポイントに該当する制御モデル上の位置を決定するとともに、この決定された位置をブレークポイントとして制御モデル上に設定するようにシミュレーション手段２０に指示する。

【００７０】

そして、ステップＳ３０３、Ｓ３０４の処理が終了した場合やステップＳ３０２にてブレークポイントの設定がなされていない旨判断された場合には、この一連の処理を終了する。なお、この一連の処理を、所定周期で繰り返し実行する代わりに、シミュレーション手段２０に制御モデルが取り込まれるとともに、プログラム実行手段２２に制御プログラムが取り込まれることをトリガとして行うようにすることがより望ましい。

【００７１】

ちなみに、このブレークポイントは、先の図５に示した制御モデルにおいては、例えばブロックＢ３とブロックＢ５とにそれぞれ設けるようにすればよい。また、最終的な出力となるブロックＢ７にのみ設けるようにしてもよい。

【００７２】

次に、上記シミュレーション手段２０による制御モデルのシミュレーションにかかる処理について図８を用いて説明する。

同図８は、制御モデルのシミュレーションにかかる処理の手順を示すフローチャートである。

【００７３】

この一連の処理においては、まずステップＳ３１１において、モデル格納部１

０に格納されている制御モデルを取り込む。次に、ステップＳ３１２、Ｓ３１３において、制御モデルの所定箇所にブレークポイントを設定する。すなわち、このステップＳ３１２、Ｓ３１３においては、上記ブレークポイント設定手段２４の指示によりブレークポイントを設定するか、先の図７に示す処理にて同期手段２６の指示によりブレークポイントを設定するかのいずれかの処理を行う。

【００７４】

続くステップＳ３１４においては、上記同期手段２６からシミュレーションの実行の指示があったか否かを判断する。そして、ステップＳ３１４にて実行の指示があった旨判断されると、ステップＳ３１５にて制御モデルのシミュレーションを実行する。このシミュレーションは、制御対象となる信号を制御モデルへの入力としてシミュレーション手段２０によって与えつつ同シミュレーション手段２０によって制御モデルを動作させることで同制御モデルによりこの入力を処理させることで行われる。

【００７５】

このシミュレーションは、制御モデルの全ての実行が終了するまで（ステップＳ３１６）、又は、ブレークポイントを検出するまで（ステップＳ３１７）行われる。そして、ステップＳ３１７にてブレークポイントが検出されると、シミュレーションを停止させるとともに、ステップＳ３１８において例えばシミュレーション手段２０の備える表示器等、出力手段を通じて外部にシミュレーションを停止している旨を通知する。

【００７６】

このようにブレークポイントにてシミュレーションを停止すると、上記同期手段２６から新たな実行の指示（ステップＳ３１４）、又は同期手段２６からの異常通知（ステップＳ３１９）を待機することとなる。そして、新たな実行指示があった場合には、ステップＳ３１５以降の処理を行うことで、制御モデルのシミュレーションを再開する。

【００７７】

一方、ステップＳ３１９において同期手段２６からの異常通知があった場合にはこの一連の処理を終了する。なお、この際、シミュレーション手段２０は異常

通知のあったときの停止位置の情報を有しているため、これを同シミュレーション手段２０の備える表示器等、出力手段を通じて外部に出力することが望ましい。また、ステップＳ３１６において、全てのシミュレーションが完了した場合には、ステップＳ３２０において例えばシミュレーション手段２０の備える表示器等、出力手段を通じて外部にシミュレーションが完了した旨を通知し、この一連の処理を終了する。

【００７８】

次に、上記プログラム実行手段２２による制御プログラムの実行にかかる処理について図９を用いて説明する。

同図９は、制御プログラムの実行にかかる処理の手順を示すフローチャートである。

【００７９】

この一連の処理においては、まずステップＳ３３１において、プログラム格納部１４に格納されている制御プログラムを取り込む。次に、ステップＳ３３２、Ｓ３３３において、制御プログラムの所定箇所にブレークポイントを設定する。すなわち、このステップＳ３３２、Ｓ３３３においては、上記ブレークポイント設定手段２４の指示によりブレークポイントを設定するか、先の図７に示す処理にて同期手段２６の指示によりブレークポイントを設定するかのいずれかの処理を行う。

【００８０】

続くステップＳ３３４においては、上記同期手段２６から制御プログラムの実行の指示があったか否かを判断する。そして、ステップＳ３３４にて実行の指示があった旨判断されると、ステップＳ３３５にて制御プログラムを実行する。このプログラムの実行は、制御対象となる信号を制御プログラムへの入力としてプログラム実行手段２２によって与えつつ同プログラム実行手段２２によって制御プログラムを動作させることで同制御プログラムによりこの入力を処理させることで行われる。

【００８１】

このプログラムの実行は、制御プログラムの全てを実行するまで（ステップＳ

３３６）、又は、ブレークポイントを検出するまで（ステップＳ３３７）行われる。そして、ステップＳ３３７にてブレークポイントが検出されると、制御プログラムの実行を停止させるとともに、ステップＳ３３８において例えばプログラム実行手段２２の備える表示器等、出力手段を通じて外部に制御プログラムの実行を停止している旨を通知する。

【００８２】

このようにブレークポイントにて制御プログラムの実行を停止すると、上記同期手段２６から新たな実行の指示（ステップＳ３３４）、又は同期手段２６からの異常通知（ステップＳ３３９）を待機することとなる。そして、新たな実行指示があった場合には、ステップＳ３３５以降の処理を行うことで、制御プログラムの実行を再開する。

【００８３】

一方、ステップＳ３３９において同期手段２６からの異常通知があった場合にはこの一連の処理を終了する。なお、この際、プログラム実行手段２２は異常通知のあったときの停止位置の情報を有しているため、これを同プログラム実行手段２２の備える表示器等、出力手段を通じて外部に出力することが望ましい。また、ステップＳ３３６において、全ての制御プログラムの実行が完了した場合には、ステップＳ３４０において例えばプログラム実行手段２２の備える表示器等、出力手段を通じて外部に制御プログラムの実行が完了した旨を通知し、この一連の処理を終了する。

【００８４】

次に、上記同期手段２６により行われる制御モデル及び制御プログラムの異常の有無の検査にかかる処理について図１０を用いて説明する。

同図１０は、上記検査にかかる処理の手順を示すフローチャートである。この処理は、先の図７に示す処理が終了したことを前提として、換言すれば、制御モデル及び制御プログラムの双方にブレークポイントが設定されたことを前提として行われる。

【００８５】

この一連の処理においては、まずステップＳ３５１において、制御モデルのシ

ミュレーションを実行するように上記シミュレーション手段２０に指示すると共に、制御プログラムを実行するように上記プログラム実行手段２２に指示する。これら制御モデルのシミュレーションと制御プログラムの実行は、上記シミュレーション手段２０により制御モデルが停止されて且つ上記プログラム実行手段２２により制御プログラムが停止されるまでの間行われる。なお、この際、制御モデルのシミュレーションと制御プログラムの実行とを同一の条件で行うべく、制御対象となる信号は、換言すれば、制御モデル及び制御プログラムによって処理される入力は同一とする。

【００８６】

そして、ステップＳ３５２において、制御モデル及び制御プログラムの双方が停止されていると判断され、且つステップＳ３５３においてこれら制御モデル及び制御プログラムの双方が実行完了していないと判断されると、ステップＳ３５４以降の処理において、上記検査を行う。

【００８７】

すなわち、ステップＳ３５４では、上記実行位置対応情報に基づき、制御モデルの停止位置と、制御プログラムの停止位置とが互いに対応する位置となっているか否かを確認する。そして、互いに対応する位置となっていないと判断されると（ステップＳ３５５）、ステップＳ３５６において上記表示手段２８を通じて制御モデルと制御プログラムとの停止位置が互いに一致しない旨（「停止位置異常」）を表示する。

【００８８】

一方、ステップＳ３５５において停止位置が一致すると判断されると、ステップＳ３５７に移行する。このステップＳ３５７では、上記変数対応情報に基づき、制御モデル側の当該停止箇所のブロック結線やストレージと、制御プログラム側の当該停止箇所において保持されている変数とが互いに対応するものであるか否かを判断する。そして、互いに対応するものでないと判断されると（ステップＳ３５８）、ステップＳ３５９において上記表示手段２８を通じて制御モデルのブロック結線やストレージと制御プログラムの変数値とが互いに対応するものでない旨（「変数値異常」）を表示する。

【００８９】

ちなみに、上記制御モデルのブロック結線やストレージと制御プログラムの変数値とが互いに対応するものであるか否かは、これら２つの値の差としての許容範囲を予め定めておき、この許容範囲にあるか否かで判断するようにする。すなわち、例えば制御モデルのブロック結線やストレージと制御プログラムの変数値とが条件である場合には、制御モデルの条件が制御プログラムの条件を全て含む場合を許容範囲とする。また例えば制御モデルのブロック結線やストレージと制御プログラムの変数値とが互いに整数演算された値である場合には、両者の一致のみを許容範囲とする。更に例えば制御モデルのブロック結線やストレージと制御プログラムの変数値とが互いに浮動小数点型の値である場合には、両者の差の絶対値が所定値以下であることを許容範囲とする。

【００９０】

上記ステップＳ３５８において、変数が許容範囲にあると判断されたときには、ステップＳ３５１に戻り、制御モデルのシミュレーションと制御プログラムの実行とが再開されることとなる。また、ステップＳ３５３において、制御モデルと制御プログラムとが全て実行されたと判断されると、ステップＳ３６０において、上記表示手段２８を通じてその旨（「全て実行完了」）を表示する。

【００９１】

一方、上記ステップＳ３５６、Ｓ３５９、Ｓ３６０の各処理が終了すると、この一連の処理を終了する。なお、上記ステップＳ３５３において制御モデルと制御プログラムとが全て実行されたと判断されたときには、ステップＳ３６０の処理に移行するに先立って、ステップＳ３５７等の処理を行うようにしてもよい。

【００９２】

このように本実施形態では、上記同期手段２６（比較手段）により制御モデルの停止位置と、制御プログラムの停止位置とが互いに対応する位置となっているか否かを確認するようにした。これにより、制御モデルの指定する複数の処理について、上記自動コード生成手段１２によって生成されたコードに新たに実行順序が付与されたとき、これが適切でない場合にはこれを検出することができる。

【００９３】

また、処理順序に誤りがある場合や生成されたコード自体に誤りがある場合には、図１０の処理により、変数値の違いを検出することができる。

このように、本実施形態では、生成されるコード含む制御プログラムのデバッグ作業において問題が検出されたとき、制御モデルの位置及び同制御モデルでの処理状態（ブロック結線やストレージ）との対応付けを自動的に行うことができる。

【００９４】

以上詳述した本実施形態によれば、以下の効果が得られるようになる。

（１）制御プログラムをプログラム実行手段２２によって実行することでその異常の有無の検査を行うに際し、これを制御モデルのシミュレーションと同期させるようにすることで、これら制御モデルのシミュレーション状況と制御プログラムの実行状況とを簡易に比較することができるようになる。

【００９５】

（２）対応情報作成手段１６の作成する対応情報を用いることで、シミュレーション手段による制御モデルのシミュレーションの実行位置と実行手段による制御プログラムの実行位置とを適切に対応付けることができる。

【００９６】

（３）制御モデル及び制御プログラムのいずれか一方にブレークポイントが設定されたときに、同期手段２６により他方にもブレークポイントが自動的に設定されるようにした。このため、例えば、外部からユーザが制御モデル及び制御プログラムのうちのいずれか一方を任意に選択してブレークポイントを設定することで上記同期処理が可能となる。

【００９７】

（４）ブレークポイントを各機能ブロックに設定可能とすることで、新たに実行順序が追加される場合であれ、その異常の有無を、制御モデルと対応させつつ簡易に検査することができるようになる。

【００９８】

（５）制御モデルのシミュレーションの順序と制御プログラムのプログラム実行順序とを比較することで、この実行順序の不一致にかかる異常を自動的に検出

することができるようになる。

【００９９】

（６）比較手段によって制御モデルのシミュレーションによって生成される値と制御プログラムの実行にかかる変数値とを比較することで、これらの不一致にかかる異常を自動的に検出することができるようになる。

【０１００】

なお、上記実施形態は、以下のように変更して実施してもよい。

・シミュレーション手段２０及びプログラム実行手段２２の動作の停止箇所は、上述したものに限らない。例えば制御プログラムをプログラム実行手段２２により１行ずつ実行及び停止させつつシミュレーション手段２０をこれと同期させてもよい。これは、例えば制御プログラム側に１行ずつブレークポイントを設定するとともに、先の図７に示す処理によって制御モデルのブレークポイントを設定することで行ってもよい。

【０１０１】

・先の図１０のステップＳ３５８において変数が許容範囲にないと判断された場合、先の図８及び図９の一連の処理を終了するようにしたが、これに限らない。例えばシミュレーション手段２０によってシミュレーションされるパラメータとプログラム実行手段２２で実行される変数値の少なくとも一方を変更し、先の図８や図９に破線にて示すようにシミュレーション及びプログラムの実行を再開するようにしてもよい。

【０１０２】

・シミュレーション手段２０やプログラム実行手段２２では、必ずしも制御モデルや制御プログラムに入力信号を与えることでシミュレーション及びプログラムの実行を行うものに限らない。例えば制御プログラムの制御対象及び同対象を取り巻く環境をコンピュータ上で擬似的に生成しつつ上記制御モデルや制御プログラムの検査を行うものであってもよい。

【０１０３】

・制御モデルから自動生成されるコードとしては、Ｃ言語にて記述されるものに限らず、適宜のプログラム言語にて記述されるコードであればよい。このコー

ドが制御モデルの規定する処理について同制御モデルにおいて規定されない実行順序が新たに生成されることがあるものであるなら、このコードと制御モデルとを同期させて検査をすることは特に有効である。

【０１０４】

・上記実施形態では、制御モデルからプログラム言語にて記述されたコードに適宜プログラムを追加したものを制御プログラムとしたが、上記コードと制御プログラムとが一致してもよい。

【０１０５】

・上記対応情報作成手段１６や同期手段２６等をコンピュータとソフトウェアにて構成する代わりに、専用のハードウェアにて構成してもよい。

・制御プログラムの作成工程としては、モデルベース開発を用いるとともに、制御モデルのシミュレーションと制御プログラムの実行とを同期させつつこれらの異常の有無を検査する範囲で適宜変更してよい。

【０１０６】

・車載エンジンの制御プログラムの作成に限らず、モデルベース開発を採用して制御プログラムを作成する場合には、本発明の適用は有効である。

【図面の簡単な説明】

【図１】本実施形態にかかる制御プログラムの作成手順を示すフローチャート。

【図２】同実施形態において、制御プログラムを実行するプログラム実行手段の構成を示すブロック図。

【図３】同実施形態にかかる制御プログラムの生成及び検査を行うシステムの構成を示すブロック図。

【図４】制御モデルを例示するブロック図。

【図５】自動生成されるコードを例示する図。

【図６】上記実施形態にかかる制御モデルと制御プログラムとの対応関係を表す対応情報を例示する図。

【図７】同実施形態におけるブレークポイントの設定にかかる処理の手順を示すフローチャート。

【図８】同実施形態にかかるシミュレーション手段による処理の手順を示すフローチャート。

【図９】同実施形態にかかるプログラム実行手段による処理の手順を示すフローチャート。

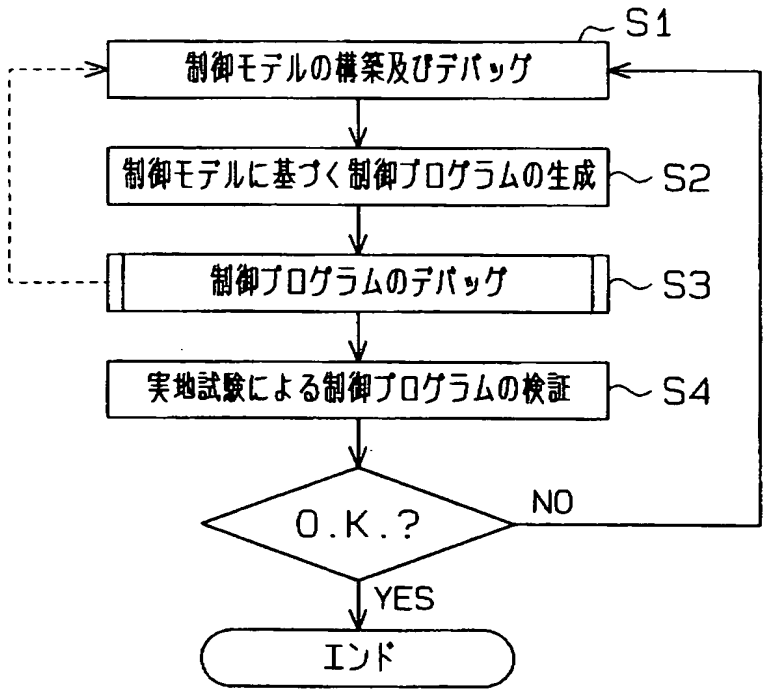
【図１０】同実施形態にかかる同期手段による処理の手順を示すフローチャート。

【符号の説明】

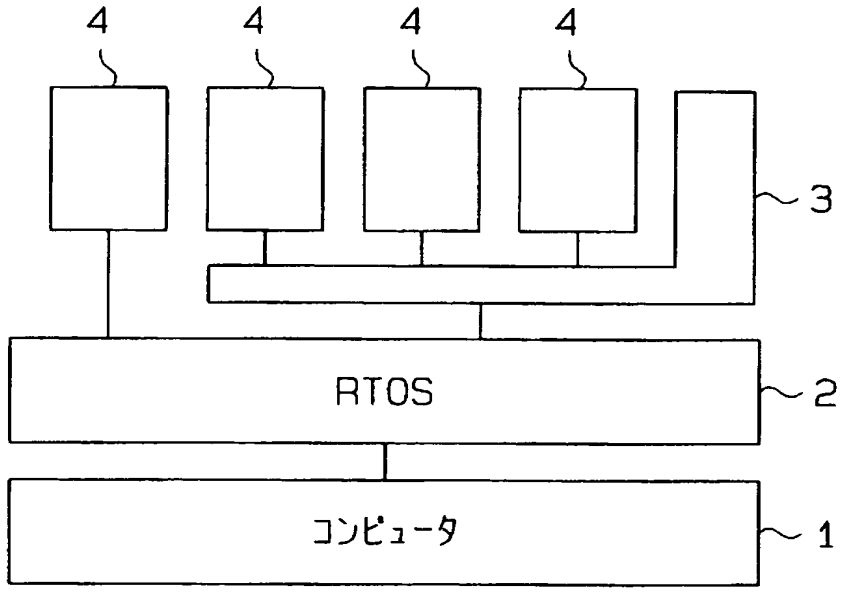
１…コンピュータ、２…リアルタイムオペレーティングシステム（ＲＴＯＳ）、３…追加プログラム、４…コード、１６…対応情報作成手段、２０…シミュレーション手段、２２…プログラム実行手段、２６…同期手段。

【書類名】　図面

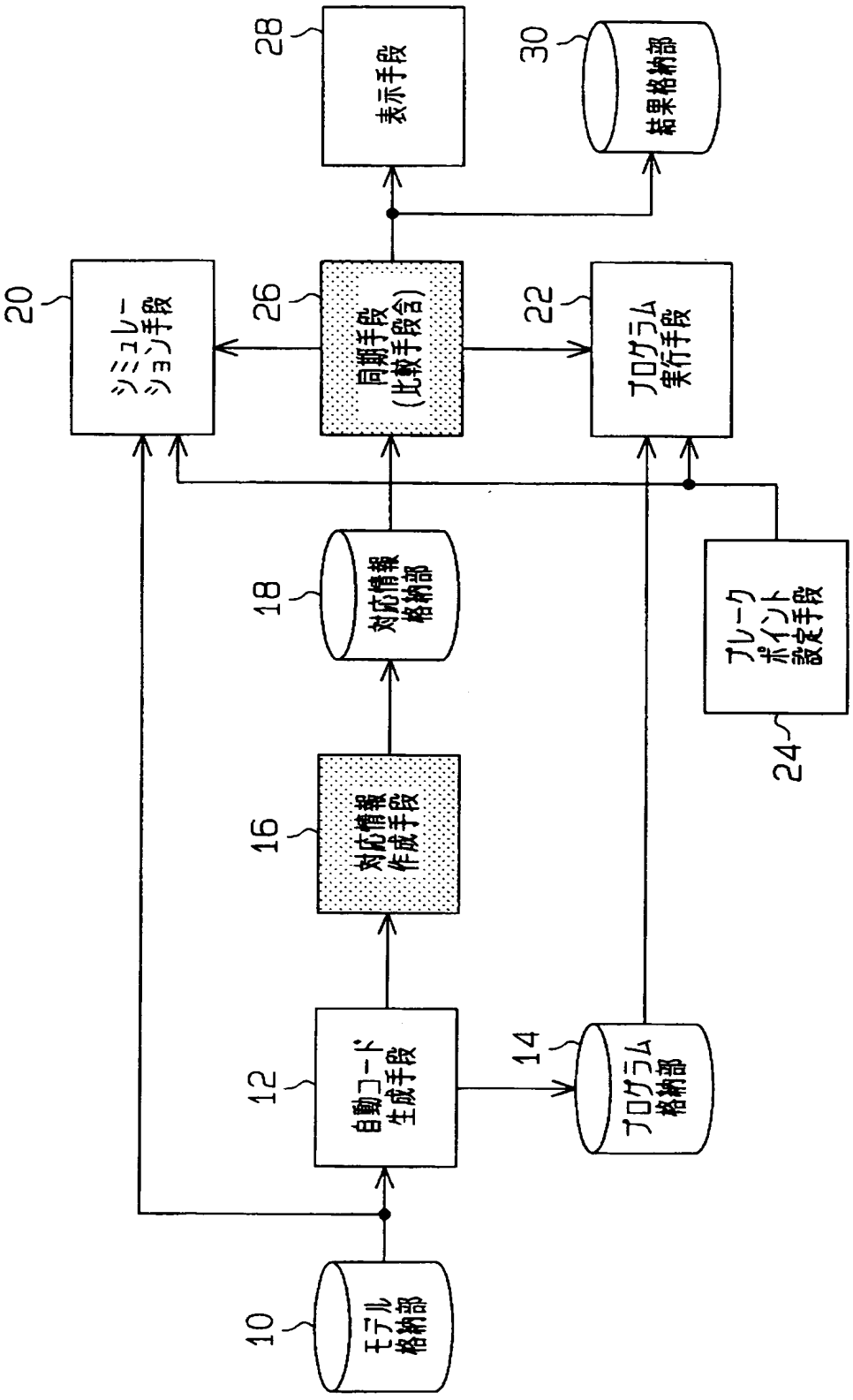
【図１】


S1
制御モデルの構築及びデバッグ
制御モデルに基づく制御プログラムの生成
S2
制御プログラムのデバッグ
S3
実地試験による制御プログラムの検証
S4
O.K.?
NO
YES
エンド

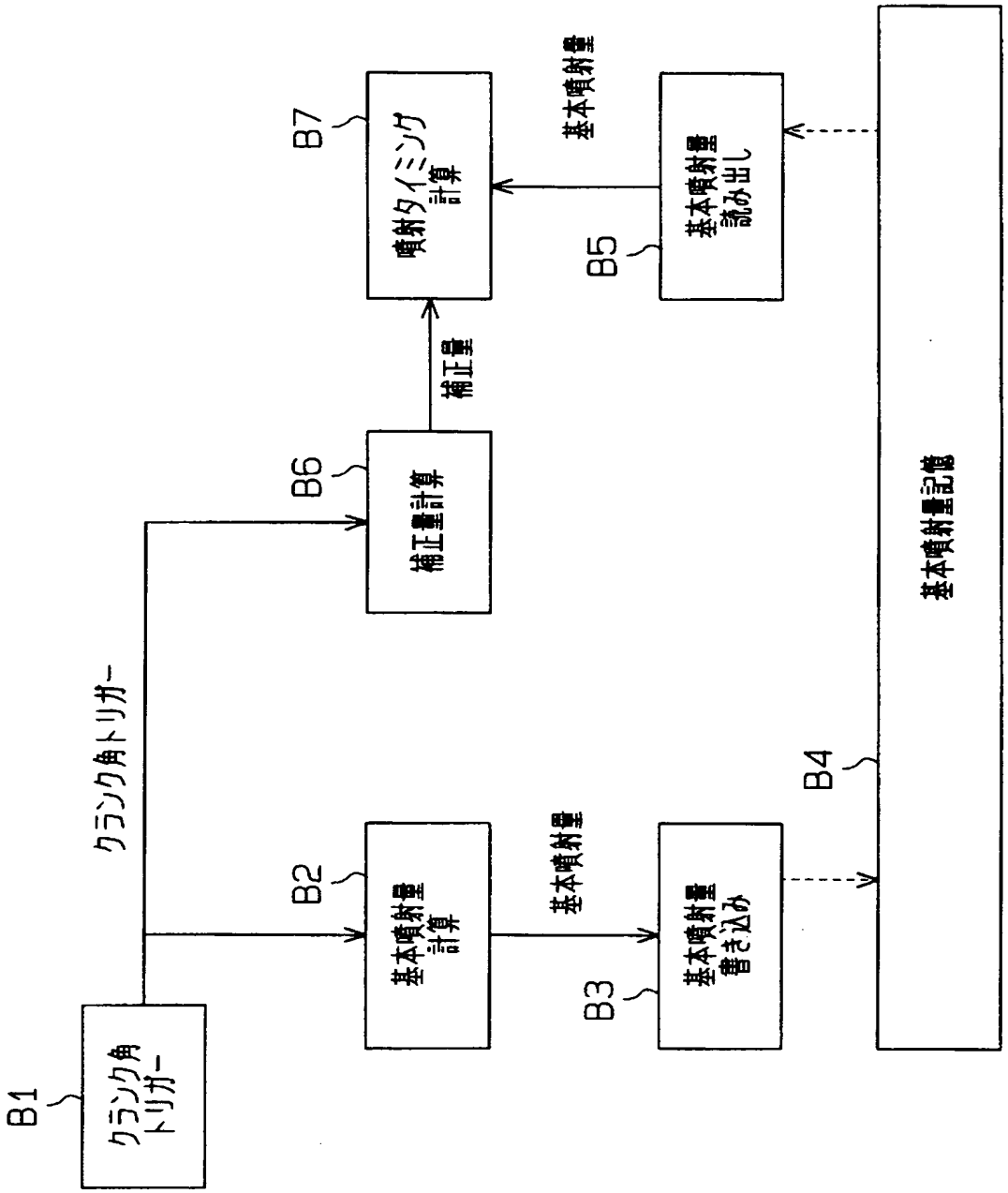
【図２】


4
4
4
4
3
RTOS
2
コンピュータ
1

【図３】


28
表示手段
30
結果格納部
20
シミュレーション手段
26
同期手段（比較手段含）
22
プログラム実行手段
18
対応情報格納部
ブレークポイント設定手段
24
16
対応情報作成手段
12
自動コード生成手段
14
プログラム格納部
10
モデル格納部

【図４】


B1
クランク角トリガー
クランク角トリガー
B2
基本噴射量計算
基本噴射量
B3
基本噴射量書き込み
B4
基本噴射量記憶
B5
基本噴射量読み出し
基本噴射量
B6
補正量計算
補正量
B7
噴射タイミング計算

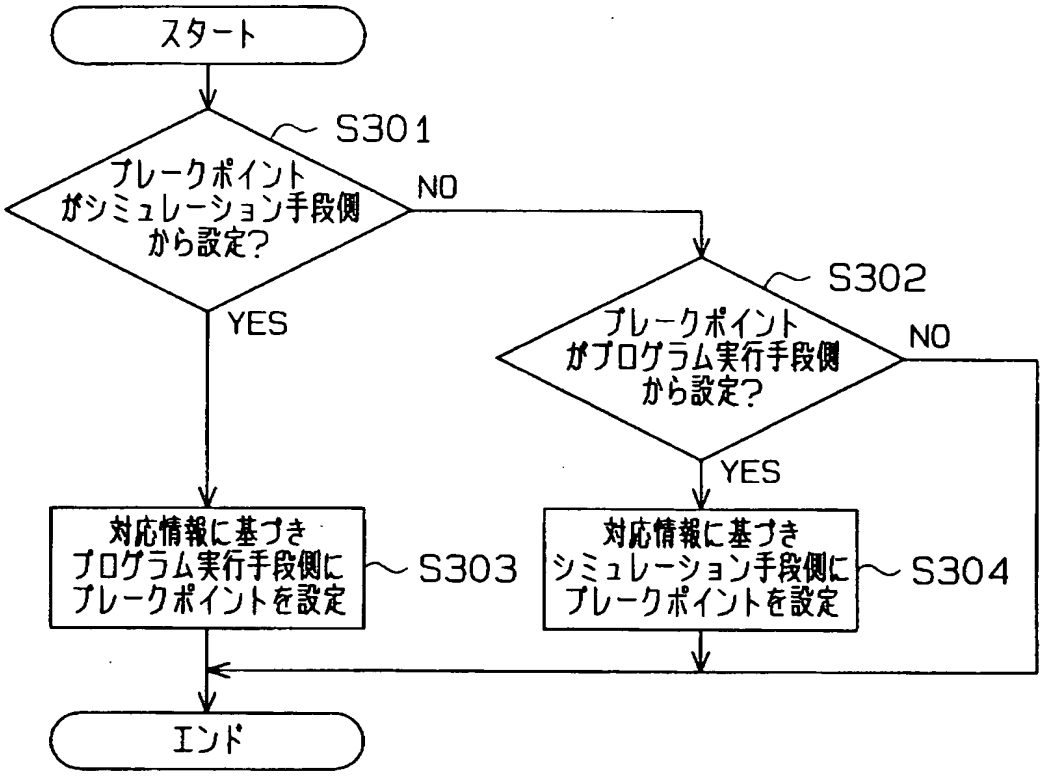
【図５】

```
L1:
   //基本噴射量計算 (B2)
   v1←...;
L3:
   //補正量計算 (B6)
   v2←...;
L4:
   v3←read (m1);  //基本噴射量読み出し (B5)
L5:
   //噴射量タイミング計算 (B7)
   v4←... (v2, v3);
L2:
   write (m1, v1);  //基本噴射量書き込み (B3)
```

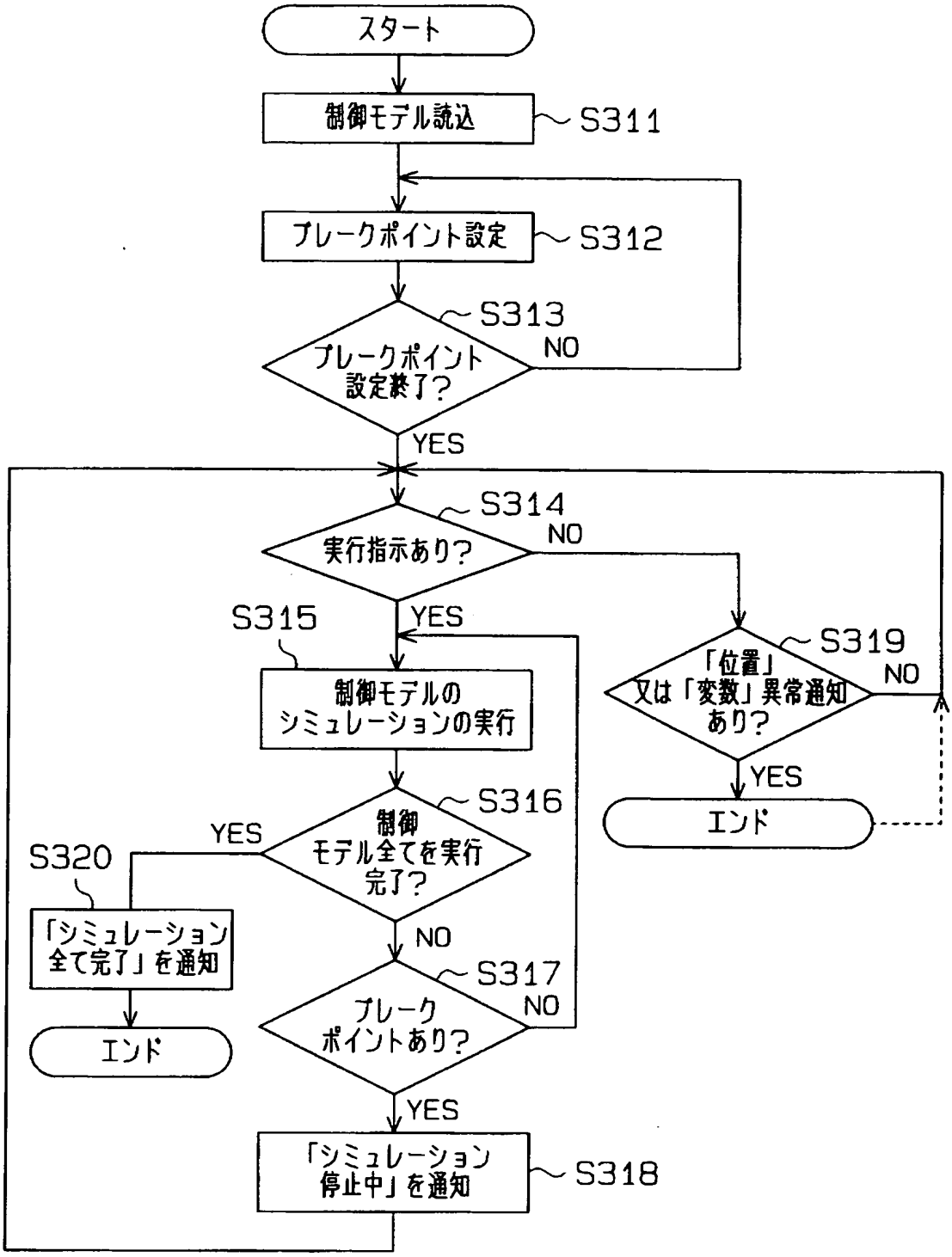
【図６】

実行位置対応情報		変数対応情報	
ブロック	ラベル	ブロック結線/ストレージ	変数
クランク角トリガー (B1)			
基本噴射量計算 (B2)	L1	基本噴射量	v1
基本噴射量書き込み (B3)	L2	基本噴射量	v1
		基本噴射量記憶	m1
補正量計算 (B6)	L3	補正量	v2
基本噴射量読み出し (B5)	L4	基本噴射量	v3
		基本噴射量記憶	m1
噴射量タイミング計算 (B7)	L5	補正量	v2
		基本噴射量	v3
		噴射量タイミング	v4

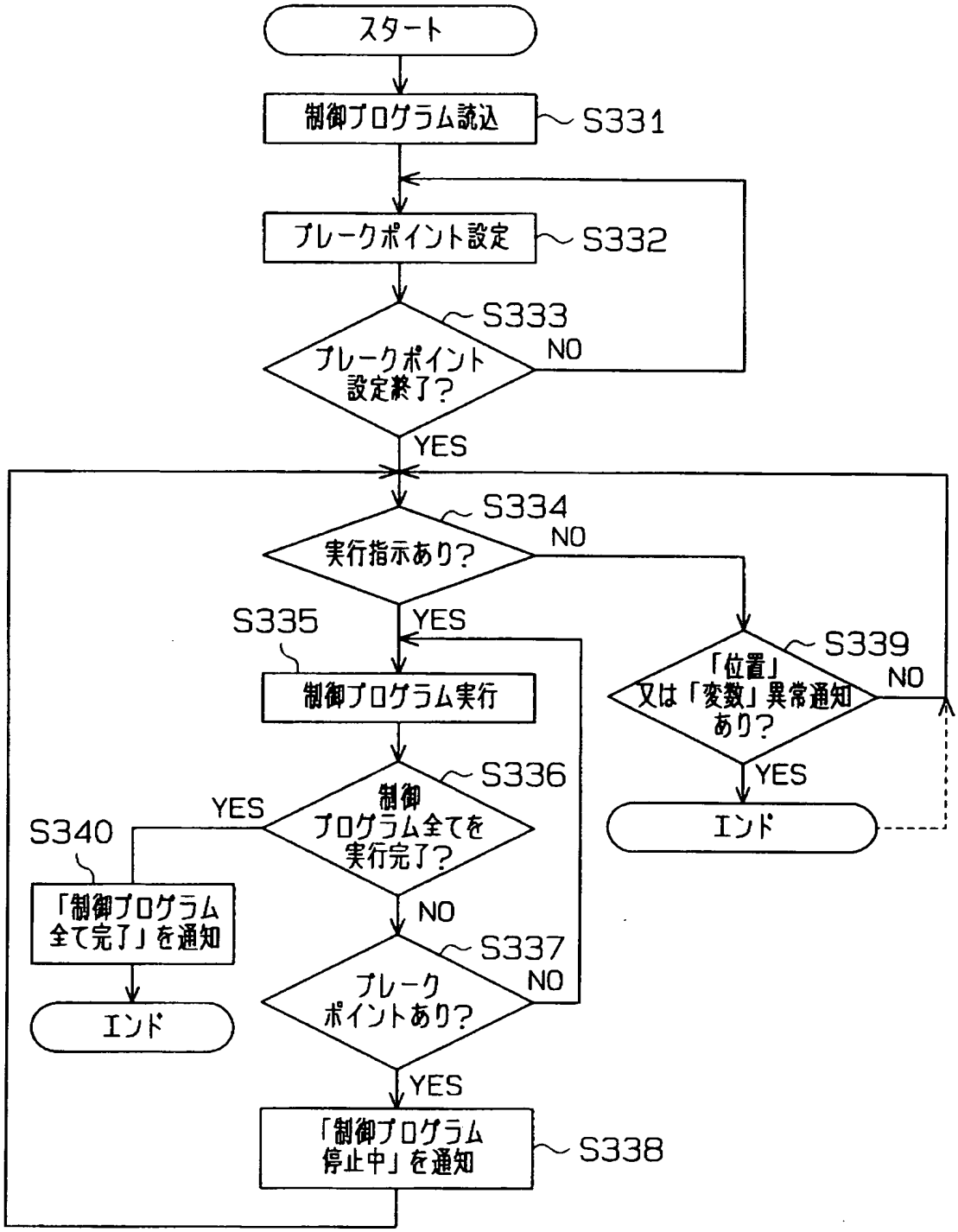
【図７】


スタート
S301
ブレークポイント
がシミュレーション手段側
から設定?
NO
YES
S302
ブレークポイント
がプログラム実行手段側
から設定?
NO
YES
対応情報に基づき
プログラム実行手段側に
ブレークポイントを設定
S303
対応情報に基づき
シミュレーション手段側に
ブレークポイントを設定
S304
エンド

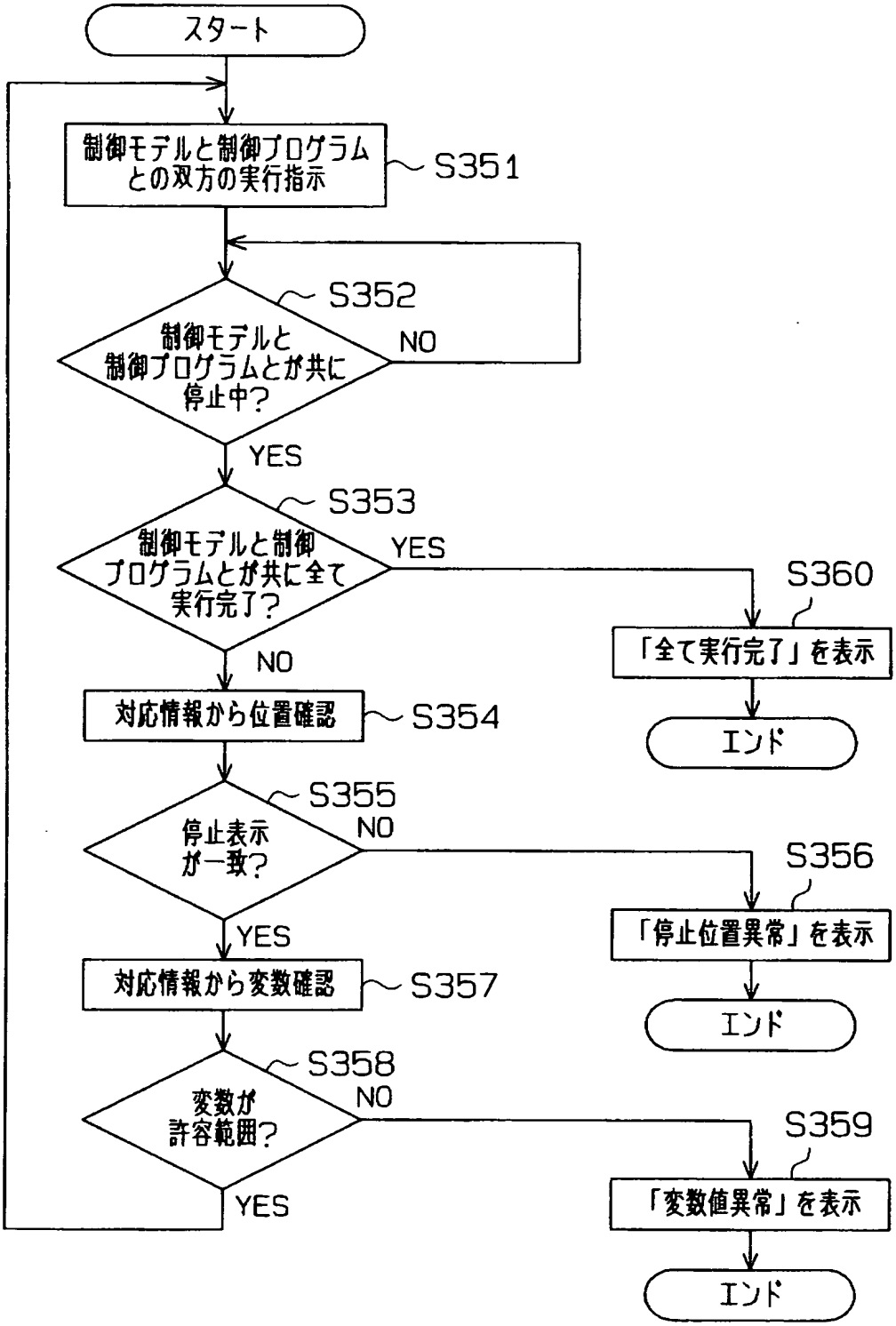
【図８】


スタート
制御モデル読込
S311
ブレークポイント設定
S312
S313
ブレークポイント設定終了?
NO
YES
S314
実行指示あり?
NO
YES
S315
制御モデルのシミュレーションの実行
S316
制御モデル全てを実行完了?
YES
S320
「シミュレーション全て完了」を通知
エンド
NO
S317
ブレークポイントあり?
NO
YES
「シミュレーション停止中」を通知
S318
S319
「位置」又は「変数」異常通知あり?
NO
YES
エンド

【図９】


スタート
制御プログラム読込
S331
ブレークポイント設定
S332
S333
ブレークポイント
設定終了?
NO
YES
S334
実行指示あり?
NO
YES
S335
制御プログラム実行
S336
制御
プログラム全てを
実行完了?
YES
S340
「制御プログラム
全て完了」を通知
エンド
NO
S337
ブレーク
ポイントあり?
NO
YES
「制御プログラム
停止中」を通知
S338
S339
「位置」
又は「変数」異常通知
あり?
NO
YES
エンド

【図１０】


スタート
制御モデルと制御プログラムとの双方の実行指示
S351
S352
制御モデルと制御プログラムとが共に停止中?
NO
YES
S353
制御モデルと制御プログラムとが共に全て実行完了?
YES
S360
「全て実行完了」を表示
エンド
NO
対応情報から位置確認
S354
S355
停止表示が一致?
NO
S356
「停止位置異常」を表示
エンド
YES
対応情報から変数確認
S357
S358
変数が許容範囲?
NO
S359
「変数値異常」を表示
エンド
YES

【書類名】　　要約書

【要約】

【課題】モデルベース開発を行う場合にあって、制御プログラムや制御モデルの異常の有無の検査を簡易に行う。

【解決手段】自動コード生成手段１２は、制御モデルを入力とし、これからＣ言語で記述されたコードを生成して出力する。対応情報作成手段１６は、上記自動コード生成手段１２がコードを生成するに際して同自動コード生成手段１２の有する情報に基づき、制御モデルとコードとの対応関係を表す対応情報を作成する。シミュレーション手段２０は、制御モデルを入力とし、その動作をシミュレーションする。また、プログラム実行手段２２は、上記コードに基づく制御プログラムを入力とし、これを実行するものである。同期手段２６は、上記対応情報に基づきシミュレーション手段２０及びプログラム実行手段２２の動作を同期させる。

【選択図】　　図３

特願２００３－０５４０８２

出　願　人　履　歴　情　報

識別番号　［０００００４２６０］

１．変更年月日　１９９６年１０月　８日
［変更理由］　名称変更
住　所　愛知県刈谷市昭和町１丁目１番地
氏　名　株式会社デンソー